Trace to Follow, Run to Explore

A Demonstration using Interactive Sorting

Pranav Vats*
vats@disroot.org
International Institute of Information Technology
Hyderabad, Telangana, India

Venkatesh Choppella
venkatesh.choppella@iiit.ac.in
International Institute of Information Technology
Hyderabad, Telangana, India

ABSTRACT

The fixed control structure of deterministic algorithms renders their behavior traceable but not amenable to interactive exploration.

In the accompanying poster, we illustrate how the runs of a transition system simulating a family of algorithms form the basis for interaction and exploration. A student interacting with a transition system controls its execution. They are running a system, not merely tracing an algorithm. The runs are easy and fun to construct as pen-and-paper based classroom activities or as interactive online simulations. We illustrate the versatility of this approach for a class of swap-based sorting algorithms and other related problems.

CCS CONCEPTS

• **Social and professional topics** → **Computational thinking**; **Computer science education**; • **Theory of computation** → **Sorting and searching**.

KEYWORDS

Interaction, Exploration, Algorithms, Transition Systems, Computing Education, Sorting

ACM Reference Format:
Pranav Vats and Venkatesh Choppella. 2023. Trace to Follow, Run to Explore: A Demonstration using Interactive Sorting. In *Proceedings of the ACM Conference on Global Computing Education Vol 2 (CompEd 2023), December 5–9, 2023, Hyderabad, India.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3617650.3624930

1 EXPLORING ALGORITHMS INTERACTIVELY VIA RUNS

A student tasked to understand an algorithm follows its execution as a sequence of observations at each step, i.e., as a *trace*. A trace unravels the control structure inherent in the algorithm, which is crucial for understanding it. The fixed nature of that control structure, however, deprives the student of the opportunity to explore alternative problem solving strategies, which—we maintain—is just as crucial. To explore, the student needs to take control of the execution of the algorithm at each step. Prior work establishes the use of Labelled Transition systems (LTS) for controlled exploration of strategies [1] by providing a choice of *actions* at each step which affords us the needed flexibility in the control structure. The operative notion there is that of a *run*—a sequence of states and actions—which, unlike a trace, explicitly incorporates the actions that drive the exploration. As a pedagogical device, an LTS may form the basis for the design of interactive notional machines which are unimpeded by programming language constructs, focusing solely on the machine to be controlled [2, §4.5, pp. 31], while still allowing for abstraction.

*Both authors contributed equally to this research.

CompEd 2023, December 5–9, 2023, Hyderabad, India

ACM ISBN 979-8-4007-0374-4/23/12.
https://doi.org/10.1145/3617650.3624930

2 EXAMPLES: INTERACTIVE SORTING

We use the simple Swap Machine [1] to explore and compare various sorting strategies. The swap machine's transition relation is expressed as a single rule: $a \xrightarrow{\mathsf{swap}(i,j)} a'$ iff $a' = swap(a,i,j)$. Here, $\mathsf{swap}(i,j)$ is the action applied to an array a and $swap(a,i,j)$ is the resultant array.

1. $[7,9,6,8] \xrightarrow{\mathsf{swap}(0,2)} [6,9,7,8] \xrightarrow{\mathsf{swap}(1,3)} [6,8,7,9] \xrightarrow{\mathsf{swap}(1,2)} [6,7,8,9]$

2. $[7,9,6,8] \xrightarrow{\mathsf{swap}(1,2)} [7,6,9,8] \xrightarrow{\mathsf{swap}(2,3)} [7,6,8,9] \xrightarrow{\mathsf{swap}(0,1)} [6,7,8,9]$

3. $$\begin{array}{ccccccc} [7,9,6,8] & \xrightarrow{\mathsf{swap}(0,1)} & [9,7,6,8] & \xrightarrow{\mathsf{swap}(1,3)} & [9,8,6,7] & \xrightarrow{\mathsf{swap}(0,3)} & [7,8,6,9] \\ & & & & & & \downarrow{\scriptstyle \mathsf{swap}(0,1)} \\ [6,7,8,9] & \xleftarrow{\mathsf{swap}(0,1)} & [7,6,8,9] & \xleftarrow{\mathsf{swap}(0,1)} & [6,7,8,9] & \xleftarrow{\mathsf{swap}(0,2)} & [8,7,6,9] \end{array}$$

Figure 1: Pen-and-paper presentations of the Swap Machine runs: 1. Ad hoc sorting, 2. Bubble sort, 3. Heap sort.

Figure 1 illustrates the versatility of an LTS, allowing the exploration of multiple strategies for sorting. In the poster, we present LTSs for exploring some swap based sorting algorithms with screenshots of their web simulations.

REFERENCES

[1] Venkatesh Choppella, Kasturi Viswanath, and Mrityunjay Kumar. 2021. Algodynamics: Algorithms as systems. In *2021 IEEE Frontiers in Education Conference (FIE)*. 1–9. https://doi.org/10.1109/FIE49875.2021.9637441

[2] Sally Fincher, Johan Jeuring, Craig S. Miller, Peter Donaldson, Benedict du Boulay, Matthias Hauswirth, Arto Hellas, Felienne Hermans, Colleen Lewis, Andreas Mühling, Janice L. Pearce, and Andrew Petersen. 2020. Notional Machines in Computing Education: The Education of Attention. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education* (Trondheim, Norway) *(ITiCSE-WGR '20)*. Association for Computing Machinery, New York, NY, USA, 21–50. https://doi.org/10.1145/3437800.3439202